

REMARKS

Applicants thank the Examiner for total consideration given the present application. Claims 31-33, 35-38, 40-43, and 45 were pending prior to the Office Action. Claims 33, 38, and 43 have been canceled and claims 46-48 have been added through this Reply. Therefore, claims 31-32, 35-37, 40-42, and 45-48 are currently pending of which claims 31, 36, and 41 are independent. Claims 31, 36, and 41 have been amended through this Reply. Upon careful review, one would conclude that the amendments made to the claims do not add any new matter to the application. Support for this amendment can be found on paragraphs [0168] to [0174] of the instant specification. Applicants respectfully request reconsideration of the rejected claims in light of the amendment and remarks presented herein, and earnestly seek timely allowance of all pending claims.

35 U.S.C. § 103 REJECTION – Jacobs, Haderle, Jenkins

A. Claims 31-33, 36-38, and 41-43 stand rejected under 35 U.S.C. § 103(a) as allegedly being unpatentable over Jacobs et al. (US 6,105,025) in view of Haderle et al. (US 4,933,848). Applicants respectfully traverse this rejection. Claims 33, 38, and 43 have been canceled through this Reply rendering the rejection of these claims as moot.

For a Section 103 rejection to be proper, a *prima facie* case of obviousness must be established. See M.P.E.P. 2142. One requirement to establish *prima facie* case of obviousness is that the prior art references, when combined, must teach or suggest all claim limitations. See M.P.E.P. 2142; M.P.E.P. 706.02(j). Thus, if the cited references fail to teach or suggest one or more elements, then the rejection is improper and must be withdrawn.

In this instance, it is respectfully submitted that neither Jacobs nor Haderle, either alone or in combination, teach or suggest all claim limitations. For example, amended independent claims 31, 36, and 41 recite, *inter alia*,

wherein the check stack contains *a list of functions that have to be executed at the end of the transaction, said functions originating from Insert, Delete and Update Data Manipulation Language*

(DML) operations calling up the stack maker module.

the Insert DML operation calling up the stack maker module leading to an insert process being performed on the check stack,

the insert process involving placing all checks that have to be executed as a result of an occurrence of a table type being inserted and corresponding conceptual rules being identified for the table type being inserted.

the Delete DML operation calling up the stack maker module leading to a delete process being performed on the check stack,

the delete process involving removing previously inserted entries on the check stack for the occurrence to be deleted and placing all checks that have to be executed as a result of a table type being deleted and corresponding conceptual rules being identified for the table type being deleted, and

the Update DML operation calling up the stack maker module *leading to said delete process followed by said insert process being performed on the check stack* (Emphasis added).

It is respectfully submitted that neither Jacobs nor Haderle, either alone or in combination, teach or suggest the above-identified features of claims 31, 36, and 41.

Jacobs is directed to a conventional method of providing a uniqueness-required index and a corresponding non-uniqueness count to support deferred uniqueness constraint enforcement wherein the non-uniqueness count associated with the uniqueness-required index provides a count of the number of duplicate entries in the index. Jacobs further discloses that the non-uniqueness counts are maintained in a list. For example, when a uniqueness-required index becomes non-unique, a uniqueness count is created for the index and added to the list and when the non-uniqueness count of any uniqueness-required index becomes zero, the non-uniqueness count is removed from the list. An empty list of non-uniqueness counts indicates that no unresolved uniqueness constraint violations remain. (See col. 4, line 61 – col. 5, line 17.) In Figs. 3A-3D, Jacobs further discloses insert, delete, and update operations using the uniqueness-required index.

The claimed invention is distinguished from Jacobs in that nowhere does Jacobs teach or suggest a check stack that contains a list of functions that have to be executed at the end of the

transaction, wherein the functions originates from Insert, Delete and Update Data Manipulation Language (DML) operations calling up the stack maker module. The claimed invention is directed to handle database transactions in general, i.e., transactions that comprise a sequence of Data Manipulation Language (DML) operations including insert, update and delete processes wherein *the insert process* involving *placing all checks that have to be executed as a result of an occurrence of a table type being inserted and corresponding conceptual rules being identified for the table type being inserted,* *the delete process* involving *removing previously inserted entries on the check stack for the occurrence to be deleted and placing all checks that have to be executed as a result of a table type being deleted and corresponding conceptual rules being identified for the table type being deleted,* and the *Update* DML operation calling up the stack maker module *leading to said delete process followed by said insert process being performed on the check stack.* As demonstrated above, Jacobs merely maintains *a list which counts the number of constraint (uniqueness) violations.* Jacobs fails to disclose a check stack having a list of functions that have to be performed at the end of the transaction associated with the specific insert, delete, and update processes mentioned above. Although Jacobs discloses a uniqueness-required index 308, such index does not include a list of functions that have to be performed at the end of the transaction as claimed corresponding to insert, delete, and update processes. The uniqueness-required index 308 of Jacobs is merely searched to verify that no duplicate entries have been made for an indexed data value.

Haderle does not fulfill at least the above-identified deficiencies of Jacobs. Haderle discloses a load utility method in which checking of data after bulk loading into a database is deferred for checking the database for violation of constraints. A bulk loading phase is an initial population process wherein the database is populated with new data, i.e., a phase in which the records of the database are loaded from an input data set into their target tables. Haderle fails to disclose a solution for handling database transactions or database manipulating operations in general, as it merely mentions constraint checks performed subsequent to a bulk loading phase i.e., a phase which includes INSERT operations only.

In contrast, as demonstrated above, the present invention is able to handle database transactions in general, i.e., transactions that comprise a sequence of DML operations including insert, update and delete operations, wherein the insert process involves placing all checks that have to be executed as a result of an occurrence of a table type being inserted and corresponding conceptual rules being identified for the table type being inserted, the delete process involves removing previously inserted entries on the check stack for the occurrence to be deleted and placing all checks that have to be executed as a result of a table type being deleted and corresponding conceptual rules being identified for the table type being deleted, and the Update DML operation calls up the stack maker module leading to said delete process followed by said insert process being performed on the check stack.

Therefore, for at least these reasons, it is respectfully submitted that independent claims 31, 36, and 41 are distinguishable over Jacobs and Haderle. Claims 32, 37, and 42 are also distinguishable over Jacobs and Haderle at least by virtue of their dependency on corresponding independent claim.

B. Claims 35, 40, and 45 stand rejected under 35 U.S.C. § 103(a) as allegedly being unpatentable over Jacobs in view of Haderle and further in view of Jenkins (US 5,899,993). Applicants respectfully traverse this rejection. Claims 35, 40, and 45 depend from claims 31, 36, and 41, respectively. As demonstrated above, the combined invention of Jacobs and Haderle fails to teach or suggest, *inter alia*, a check stack having a list of functions that have to be performed at the end of the transaction associated with the specific insert, delete, and update processes as recited in claims 31, 36, and 41. Jenkins has not been, and indeed cannot be, relied upon to fulfill at least these deficiencies of Jacobs and Haderle. Therefore, it is respectfully submitted that claims 35, 40, and 45 are at least allowable by virtue of their dependency on corresponding independent claims 31, 36, and 41, respectively.

NEW CLAIM

New claims 46-48 depend from claims 31, 36, and 41, respectively. Therefore, it is respectfully submitted that claims 46-48 are at least allowable by virtue of their dependency on

corresponding independent claims 31, 36, and 41, respectively and further in view of novel feature recited therein.

Conclusion

In view of the above amendment and remarks, it is believed that all pending claims are allowable.

Should there be any outstanding matters that need to be resolved in the present application, the Examiner is respectfully requested to contact Ali M. Imam Reg. No. 58,755 at the telephone number of the undersigned below, to conduct an interview in an effort to expedite prosecution in connection with the present application.

If necessary, the Commissioner is hereby authorized in this, concurrent, and future replies to charge payment or credit any overpayment to Deposit Account No. 02-2448 for any additional fees required under 37.C.F.R. §§1.16 or 1.14; particularly, extension of time fees.

Dated: August 18, 2008

Respectfully submitted,

By  #58,755
Michael K. Mutter
Registration No.: 29,680
BIRCH, STEWART, KOLASCH & BIRCH, LLP
8110 Gatehouse Road
Suite 100 East
P.O. Box 747
Falls Church, Virginia 22040-0747
(703) 205-8000
Attorney for Applicant